

UNCLASSIFIED

AD-A241 807



2

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY DTIC ELECTE OCT 22 1991		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR-91-0881	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6a. NAME OF PERFORMING ORGANIZATION The Pennsylvania State University		7b. ADDRESS (City, State, and ZIP Code) AFOSR/NE, Bldg 410, Bolling AFB, DC 20332-6448.	
6c. ADDRESS (City, State, and ZIP Code) Department of Computer Science University Park, PA 16802		9. PROCUREMENT INSTRUMENT ID NUMBER AFOSR 87-0400	
8a. NAME OF FUNDING SPONSORING ORGANIZATION Air Force Office of Scientific Research		10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code) AFOSR/NE, Bldg 410, Bolling AFB, DC 20332-6448.		11. TITLE (Include Security Classification) A Complexity Theory of Neural Networks (Unclassified)	
12. PERSONAL AUTHOR(S) P. Berman, I. Parberry, and G. Schnitger			
13a. TYPE OF REPORT Final Technical	13b. TIME COVERED 15 SEP-44 APR 91	14. DATE OF REPORT 1991, August 9	15. PAGE COUNT 13
16. SUPPLEMENTARY NOTATION The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Office of Scientific Research or the U.S. Government.			
17. COSATI CODES		18. SUBJECT TERMS Neural networks, complexity theory, fault tolerance, learning.	
19. ABSTRACT Significant progress has been made in laying the foundations of a complexity theory of neural networks. The fundamental complexity classes have been identified and studied. The class of problems solvable by small, shallow neural networks has been found to be the same class even if (1) probabilistic behaviour, (2) multi-valued logic, and (3) analog behaviour, are allowed (subject to certain reasonable technical assumptions). Neural networks can be made provably fault-tolerant by physically separating the summation units from the thresholding units. New results have also been obtained on the complexity of approximation, communication complexity, the complexity of learning from examples and counterexamples, learning with multi-valued neurons, exponential lower bounds for restricted neural networks, and fault tolerance in distributed computation.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Steven C. Suddarth	22b. TELEPHONE 202/767-5028	22c. OFFICE SYMBOL NE/NM	

91-13726

91 10 22 011

UNCLASSIFIED

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

A COMPLEXITY THEORY OF NEURAL NETWORKS



Piotr Berman, Georg Schnitger
 Dept. of Computer Science
 333 Whitmore Laboratory
 Penn State University
 University Park, PA 16802

Ian Parberry
 Dept. of Computer Sciences
 University of North Texas
 P.O. Box 13886
 Denton, TX 76203-3886

7 August 1991

Final Technical Report for Period September 15, 1987 to June 30, 1991

Prepared for:

Capt. S. Suddarth
 AFOSR/NE
 Building 410
 Bolling AFB, DC 20332-6448

1 Research Objectives

Complexity theory is the study of resource-bounded computation. The aim of this project is to study the amount of resources, in particular, time and hardware, used in neural network computations. Research will focus on four major topics:

1. The relative computing power of various neural network models.
2. Algorithms for neural network computations; upper-bounds, lower-bounds and completeness properties.
3. Fault-tolerant computation.
4. Learning.

2 Accomplishments

The Investigators have made significant progress in laying a foundation for a complexity theory of neural networks. The complexity class TC^0 has been identified as the prime class of interest for neural network computation. It consists of the problems which can be solved by small, fast neural networks, that is, those whose size grows only polynomially with the number of inputs, and with a fixed number of layers. The complexity class remains the same under many different neural network models, for example, even if probabilistic ([19]), multi-valued ([15]), or analog neurons are allowed. Progress has also been made with problems related to learning and fault tolerance. More details follow.

2.1 Foundations

The article by Parberry [18] has laid the groundwork for the study of the complexity of neural networks. In this paper, a case is made for the importance of the complexity theory of neural networks by comparing and contrasting it with conventional sequential, parallel and probabilistic complexity theory, and collect together much of the knowledge which can be fairly easily deduced from standard results in complexity theory. This will form the background against which our research will be presented. The key resources of *time*, *size*, (number of neurons) and *weight* (total weight of all connections) are identified. The latter two resources give some indication of the amount of hardware that will be needed to implement neural networks. The links between neural network based complexity classes and the standard classes are explored. There is no significant difference between the two until running time is reduced to a constant and hardware to a polynomial. In this case (for example, the class TC^0 of languages which can be recognized in polynomial hardware and constant depth) little is known.

Parberry [18] also contains a few previously unpublished results, most notably the following.

1. The weights of a neural network can be made ± 1 by increasing the size from z to $z^4 \log^3 z$ and the depth by a constant multiple. This is a smaller increase in size than previously obtained in Parberry and Schnitger [19].
2. Any function which can be computed by a conventional¹ circuit of size z and depth d can be computed by a neural network of depth $d/\epsilon \log \log z$ and size $O(z^{1+\epsilon})$, for any $\epsilon > 0$. This means that polynomial size neural networks are faster than conventional bounded fan-in circuits by a factor of $\log \log n$. The exponent in our result is smaller than the one previously known.
3. The NP-completeness of some problems related to the termination of cyclic neural networks has been strengthened to some restricted cases, including bounded degree, and the property that if there is a terminating computation then there is at least one which does so in polynomial time.

2.2 Pebbling

One of the advantages that neural networks have over conventional circuits is unbounded fan-in. There is a well-known relationship between size of a conventional circuit and depth of an unbounded fan-in circuit: any function computed in size z by the former can be computed in depth $O(z/\log z)$ (and possibly size exponential in z) by the latter. Kalyanasundaram and Schnitger [11] have improved this result by reducing size substantially.

2.3 Boltzmann Machines

We have formulated and developed the thesis that the class TC^0 is fundamental to neural network computations, in that it characterizes the languages recognizable by small, fast neural networks. In Parberry and Schnitger [19], we showed that this is even true for probabilistic models (such as the Boltzmann machine) of polynomial weight. Our efforts in this direction have led to an interest in TC^0 by many prominent complexity theorists.

2.4 Computing with Noisy Neurons

We consider the scenario in which each neuron has a small probability of failing, and we wish to construct a network which reliably computes the correct result with probability of some fixed constant greater than one half. The reliable simulation of fault-free conventional circuits by a faulty neural network with a log-linear increase in size and constant multiple increase in depth is described in Parberry [18]. This can be extended to neural networks with a small fixed fan-in [6]. However, the more

¹We use the term *conventional circuit* to describe a circuit constructed from two-input NAND gates.

desirable fault-tolerant simulation of a fault-free neural network appears difficult. There is evidence from VLSI and from natural neural systems that it is architecturally advantageous to physically separate the summation from the thresholding in the neuron. In [6] we propose such a model, called the *summation network*. It is not too different from the standard model, in the sense that each can simulate the other in a fault-free environment with only a polynomial increase in size and a constant multiple increase in depth (thus polynomial hardware, constant depth summation circuits also recognize TC^0). Nonetheless, summation networks are much easier to analyze in the presence of faults. We were able to obtain a reliable simulation of a fault-free summation network by a faulty summation network with a log-linear increase in size and constant multiple increase in depth.

2.5 Complexity of Approximation

It is often hoped that neural networks will be useful in solving NP-complete problems. It is apparent from [18] that no polynomial size neural network can ever solve such a problem exactly, and that it is easy to find an exponential size network which does. A much more reasonable aim would be to use neural networks to give approximate solutions, that is, solutions which are sufficiently close to the optimum. Unfortunately there is not yet available a well-developed theory of approximation algorithms. Berman and Schnitger [9] have contributed to the foundations of such a theory by investigating "approximation complete" problems. Strong evidence is provided indicating that Constraint Satisfaction problems of quite simple structure can not be approximated satisfactorily in polynomial time. Any such problem would be as difficult to approximate by a neural network as by a conventional computation [5].

2.6 Communication Complexity

If neural networks are to be implemented in VLSI, it is likely that efficient methods of solving problems in Numerical Linear Algebra are needed. The communication complexity of a function f measures the communication capacity any system computing f must provide. In the design of VLSI systems, where savings on the chip area and computation time are desired, this complexity dictates an *area \times time*² lower bound. Chu and Schnitger [10] investigate the communication complexity of determining whether a given square matrix M is singular. We show that, for $n \times n$ matrices of k -bit integers, the communication complexity of this problem is $\Theta(kn^2)$. In case the entries of M are elements of a finite field of size p , we also prove the communication complexity of this problem to be $\Theta(n^2 \log p)$. Our results imply tight bounds for a wide variety of other problems in Numerical Linear Algebra. Among those problems are determining the rank and computing the determinant of a matrix, as well as the computation of several matrix decompositions. Another important corollary concerns the solvability of linear systems. In this problem it has to be decided whether a linear system $Ax = b$ has a solution. When A is an

$n \times n$ matrix of k -bit integers and b a vector of n k -bit integers, we determine its communication complexity to be $\Theta(kn^2)$.

2.7 Learning from Tests and Counter-examples

It is hoped that neural networks will be more appropriate for learning than conventional computer models. The theory of learning has recently been the subject of much interest following the seminal contributions of Valiant.

Berman and Roos [8] have extended the work of Angluin on learning finite-state languages to show that deterministic one-counter languages (a large subset of the context-free languages) can be learned in polynomial time. In this model of learning the student (i.e. the learning algorithm) can test whether a chosen word belongs to the given language. After a sequence of such tests the student constructs a machine consistent with all tests and examples collected so far, and uses the constructed machine to predict the membership of the future examples. After an incorrect prediction the student constructs another machine with the aid of additional tests. This cycle may repeat a number of times, but the number of wrong predictions (i.e. the number of counter-examples) and the total time used for internal computations and tests is bounded by a polynomial in the number of states of the machine which recognizes the given language.

While the algorithm of Angluin always returns the minimal machine consistent with the data, the algorithm of Berman and Roos merely constructs an equivalent machine with size which is polynomially related to the minimal one. This is unavoidable, given the current state of knowledge: while deterministic finite automata can be efficiently minimized, no feasible minimization procedure is known for one-counter languages.

2.8 Multi-valued Neurons

Much experimental neural network research involves analog neurons, which input real values, and output real values. However, whilst the theory of analog neural networks developed to date uses real numbers, experimental work is typically performed on digital computers. Surprisingly, the simulations bear out the theory, even though the former is inherently discrete, and the latter inherently analog. Thus it appears that neural networks are robust in terms of precision. This is a particularly important trait, since it is impossible to fabricate analog hardware which has arbitrarily high precision. In particular, biological systems perform well with wetware which has analog behavior, but only limited precision.

The Principal Investigator, Ian Parberry, (in cooperation with his Ph.D. student Zoran Obradovic) undertook to investigate analog neural networks with limited precision. In digital simulations, the activation levels of the neurons are limited to some fixed number of values, k , which depends on the particular computer in use. The computational and learning complexity of limited precision analog neural networks was investigated, with a particular emphasis on how the number of neurons and running time scale with k , as well as the size of the problem being solved.

The key to the research was the demonstration in [15] (also submitted to *Journal of Computer and System Sciences*) that limited precision analog neural networks with k activation levels are equivalent to discrete neural networks with k levels of activation, and $k - 1$ thresholds, as opposed to the traditional single one. The computational complexity of these k -ary neural networks was studied in [15], and the learning complexity in [16,17].

The work in [15] extends the traditional binary discrete neural network complexity theory (see [18]) to the new multi-level discrete case. The reader is referred to the journal papers for details. One typical result is that unlike the binary and ternary case, the threshold values for the k -ary case where $k > 3$ cannot be fixed. For example, in the binary case, the threshold can be made 0. In the ternary case, the two thresholds can be made 0 and 1. In the general case, no fixed thresholds will suffice. If k is restricted to grow only polynomially with the size of the problem being solved, then polynomial size, constant depth k -ary neural networks compute only functions from TC^0 , the classical complexity class for binary neural networks. This implies that the superiority of analog neural networks over discrete binary ones can only confer a polynomial in size and a constant multiple in depth. However, that polynomial may still be significant.

The work in [16,17] extends the learning algorithms for the binary discrete neuron to the k -ary case. Efficient versions of the Perceptron Learning Algorithm and Littlestone's Winnow Algorithm are given, proved correct, and analyzed.

2.9 Lower Bounds for Depth 3

Ian Parberry and his student Peiyuan Yan have made some progress on lower-bounds. Whilst it is extremely difficult to obtain exponential size lower-bounds on the size required by constant depth neural networks to compute certain functions, we have made some progress by restricting the power of the neurons. We [21] consider depth 2 circuits of $mod - p$ and $mod - q$ gates augmented with the limited use of AND and OR gates with small fan-in. We are able to show an exponential size lower-bound for certain depth-3 circuits of these gates for computing Boolean conjunctions.

2.10 Computing with Analog Neurons

In [20] Georg Schnitger (in cooperation with Wolfgang Maass of the University of Illinois and Eduardo Sontag of Rutgers University) examined the computing power of feedforward networks with sigmoid (i.e. smooth) threshold gates for computing boolean functions.

A threshold gate with inputs x_1, \dots, x_n , weights w_1, \dots, w_n and threshold t outputs the real number $\gamma(\sum_{i=1}^n w_i x_i)$. Popular choices for the gate function γ include the *binary threshold function* (i.e. $\gamma(y) = 1$ if $y \geq 0$ and $\gamma(y) = 0$ if $y < 0$) and smooth threshold functions (i.e. γ is differentiable).

We demonstrate, for a large class of smooth gate functions γ (including the

standard sigmoid $\sigma(x) = \frac{1}{1 + \exp(-x)}$, that their corresponding feedforward nets are computationally at least as efficient as feedforward nets composed of binary threshold gates. Moreover, we exhibit a problem (namely to decide whether exactly one of two n -bit strings has a majority of ones) that can be solved with one hidden layer and 5 smooth gates, but the same problem cannot be solved with one hidden layer and constantly many binary threshold gates.

This raises the question whether smooth threshold gates give rise to dramatically increased computing power compared with binary threshold gates. Under quite liberal assumptions (which are satisfied by the standard sigmoid) we show that a feedforward net with n binary inputs, s smooth gates and d hidden layers can be simulated within the same number of layers by a feedforward net composed of $O(\text{poly}(n + s))$ binary threshold gates.

Thus, *disregarding* a polynomial increase in size, smooth threshold functions and binary threshold functions are computationally equivalent. If we don't disregard polynomial increases, smooth threshold functions are computationally at least as powerful as binary threshold functions and provably more powerful for certain problems.

2.11 Fault Tolerance

With his student, Mirjana Obradovic (who was supported by this grant) Piotr Berman was working on optimizing threshold gates; i.e. on minimizing the sum of the weights (assuming integer weights). When the weights are allowed to be large integers, then merely testing the equivalence of two gates is a co-NP complete problem, hence optimization cannot be feasible. However, when the sum of the weights of even one of the gates involved in the equivalence test is polynomial, then an equivalence test can in polynomial time return the confirmation of the equivalence or a counterexample. We have developed a heuristic which uses this equivalence test as follows. It maintains a set of examples for the given threshold gate, a set of proven inequalities of the form: this input should have the value of the target function at least as high as that input, and a small set of assumed inequalities. The goal is to construct a linearly ordered list of combinations of input variables, such that the minimal weight of each input is equal to its rank in the list. This work is currently continued with two graduate students, Nicol So and Ching-hoi Sze.

While this work is still in preparation, the partial results happened to have very interesting applications in the area of management of replicated data bases [13,14]. Here the subject is a data base in which data items are replicated and distributed between some number of sites, which may improve the reliability (a failure of several data sites does not render a piece of data unreachable) and access (local rather than remote reads). A static scheme allows to perform a data base transaction dependent on the set of processors which can at a particular instance of time communicate with the originator of the transaction. In a voting scheme the sets of processors allowed to execute are characterized by a distribution of votes and a quorum threshold. We

have characterized several important classes of systems in which voting schemes provide the optimal static scheme. Moreover, we introduced efficient and practical algorithms to compute the optimal distribution of votes. A part of our technique is an efficient test for the equivalence of threshold gates.

With his former student, Juan A. Garay (now at IBM T.J. Watson Research Center) Berman continued investigations on the Distributed Consensus problem. In this problem a group of processors has the task of reaching a common decision. Each processor has its initial option (typically, a 0/1 value) and the common decision must be consistent with the initial option of one of the processors. There are two complications which make this problem non-trivial: the communication is conducted via bilateral links (so no 'public' vote is possible) and some of the processors are faulty. No assumptions whatsoever are placed on the behavior of the faulty processors, e.g. they could be controlled by an omniscient adversary.

The goal of our research was to provide solutions with better quality parameters than the previous ones. The parameters which we study are the following: the resiliency, i.e. the tolerated number of faulty processors, the number of communication rounds and the amount of communication. So far, we do not know any solution which would be superior simultaneously in all these aspects. We found a solution which uses 1 bit messages, and has asymptotically optimal resiliency ($3/4$ of the optimum) and number of rounds (2 times optimum). In collaboration with K.J. Perry of IBM Watson we found a solution which has optimal resiliency, while the message size is limited to 2 bits and the number of exchange rounds is 3 times larger than the optimal one. In both cases we can substantially reduce the number of rounds by increasing message size to a higher constant (this is quite important in practice, since the cost of sending one-page message and one-bit message is usually the same). Both protocols have the form of a simple sequence of votes, in the second protocol there is a possibility of casting an undecided vote (hence 2 bits in a message, rather than 1). These results and their applications are the subject of [3,4].

Another group of results concerned protocols with optimal (rather than near optimal) number of rounds and relatively small (so-called polynomial) message size. One of these results was presented at FOCS and is the subject of the paper invited to a Special Issue of the journal *Mathematical Systems Theory* [2]. Another is the subject of [1]. While these results are also based on voting, the votes are nested recursively, which could easily lead to huge message size. The techniques developed by Berman and Garay allow a processor to avoid participation in most of possible votes, hereby reducing the message size. In particular, a set of rules was found which allow to identify quickly the faulty processors that "harm" the computation and to deduce the outcomes of avoidable votes.

The experience gained in the work on Distributed Agreement allowed us to obtain some interesting results on fault diagnosis for multiprocessor distributed systems (in cooperation with Andrzej Pelc of the University of Quebec [7]). In the fault diagnosis model we assume that the faulty processors compute unreliably, and can alter the content of transmitted messages, however they can be detected by their network neighbors with some probability; moreover faulty processors form a

random subset of the system. The previous diagnosis technique was based on a simple threshold: the processors are diagnosed to be faulty based on the number of "failed tests" (a good processor may fail a test, if the latter is "administered" by a faulty one). We have shown that the quality of diagnosis improves substantially if we form a graph of processors, and solve a maximum independent set problem for this graph (an arc is introduced between two processors whenever one of them claims that the other has failed its test). While the maximum independent set problem is in general not feasible, we have shown that it suffices to form a collection of very small graphs, and tackle them separately. Moreover, we have exhibited a scheme which allows to distribute the test result reliably through the system even with a very small number of connections (if we have n processors, then the number of links and tests is of the order $n \log n$, we have proven that this order of growth is sufficient and necessary).

3 Personnel

Dr. I. Parberry served as Principal Investigator from the inception of this grant until his resignation from Penn State University on June 30, 1990. Dr. Schnitger was Co-PI up to that date, and from then on took over as Principal Investigator. Dr. Piotr Berman was Co-Principal Investigator for the entire grant period.

Research Assistants at various times during the grant period include the following:

- Balasubramanian Kalyanasundaram. Ph. D. granted 1988. Ph. D. thesis titled "Lower Bounds on Communication, Space and Time".
- Pei Yuan Yan. Ph. D. granted June 1989. Ph. D. thesis titled "Lower Bound Techniques in Some Parallel Models of Computation".
- Mirjana Obradovic. Ph. D. granted May 1991. Ph. D. thesis titled "Management of Replicated Data in Distributed Systems".
- Zoran Obradovic. Ph. D. granted February 1991. Ph. D. thesis titled "Discrete Multi-Valued Neural Networks".
- Toshihiro Fujito. Ph. D. candidate in Computer Science.

4 Oral Presentations

- P. Berman, "Improved Algorithm for the Distributed Consensus Problem", Colloquium, Computer Science Department, University of Southern California (July 1988).
- P. Berman, "Learning One-counter Automata", Colloquium, Computer Science Department, University of California at San Diego (July 1988).
- P. Berman, "Learning One-counter Automata", Colloquium, T.J. Watson I.B.M. Research Center (August 1988).
- P. Berman, I. Parberry and G. Schnitger, "The Complexity of Reliability and Constraint Satisfaction in Neural Networks". International Neural Network Society First Annual Meeting, Boston, MA, Sept. 1988.

- I. Parberry, "A Primer on the Complexity Theory of Neural Networks". Workshop on Complexity Theory of Neural Networks, Post Meeting Workshop, IEEE Conference on Neural Information Processing Systems — Natural and Synthetic, Keystone, Colorado, Dec. 1988.
- I. Parberry, "Computing with Noisy Neurons: An Overview of Classical Techniques". Workshop on Fault Tolerance, Post Meeting Workshop, IEEE Conference on Neural Information Processing Systems — Natural and Synthetic, Keystone, Colorado, Dec. 1988
- I. Parberry, "The Complexity of Reliability in Neural Networks". Colloquium, Department of Computer Science, University of Queensland, Australia (Dec. 1988).
- I. Parberry, "Limited Precision Analog Neural Networks". Colloquium, Applied Research Laboratory, Penn State University (Sept. 1989).
- I. Parberry, "Limited Precision Analog Neural Networks". Colloquium, Department of Computer and Information Science, University of Delaware (Dec. 1989).
- P. Berman and G. Schnitger, "On the Approximation Complexity of the Independent Set Problem", 6th Annual Symposium on Theoretical Aspects of Computer Science, Paderborn, FRG (Feb. 1989).
- B. Kalyanasundaram and G. Schnitger, "Rounds versus time for the two person pebble game", 6th Annual Symposium on Theoretical Aspects of Computer Science, Paderborn, FRG (Feb. 1989).
- P. Berman, "Reliable Systolic Arrays Build of Unreliable Components", Colloquium, Computer Science Department, Carnegie-Mellon University (March 1989).
- P. Berman, "On Difficulty of Approximating Maximal Independent Set", Colloquium, Computer Science Department, University of Montreal (May 1989).
- P. Berman, "Asymptotically Optimal Distributed Consensus", Colloquium, Computer Science Department, University of Quebec at Hull (May 1989).
- J. Chu and G. Schnitger, "The Communication Complexity of Several Problems in Matrix Computation". 1st ACM Symposium on Parallel Algorithms and Architectures, Santa Fe, NM (June 1989).
- P. Berman and J. Garay, "Asymptotically Optimal Distributed Consensus", Sixteenth International Colloquium on Automata, Languages, and Programming, Stresa, Italy (July 1989).
- P. Berman and J. Garay, "Efficient Agreement on Bounded-Degree Networks". International Conference on Parallel Processing, St. Charles, Illinois (Aug. 1989).
- P. Berman, J. Garay, and K. Perry, "Towards Optimal Distributed Consensus", Thirtieth IEEE Symposium on Foundations of Computer Science, Research Triangle Park, NC (Oct. 1989).
- I. Parberry and Z. Obradovic, "Analog Neural Networks of Limited Precision I: Computing with Multilinear Threshold Functions", IEEE Conference on Neural Information Processing Systems, Natural and Synthetic, Denver, CO (Nov. 1989).
- P. Berman, "Efficient Consensus in Optimal Number of Rounds", Colloquium, Computer Science Department, Yale University (January 1990).

- P. Berman, A. Pelc, "Distributed Probabilistic Fault Diagnosis for Multiprocessor Systems", Twentieth IEEE Symposium on Fault Tolerant Computing, Newcastle upon Tyne, England (June 1990).
- Z. Obradovic and I. Parberry, "Learning with Discrete Multi-valued Neurons", Seventh Annual Machine Learning Conference, Austin, TX (June 1990).
- P. Berman, "Distributed Consensus in Bounded-degree Networks", Colloquium, University of California at Riverside (September 1990).
- P. Berman, "Distributed Consensus in Bounded-degree Networks", Colloquium, University of California at San Diego (September 1990).
- P. Berman, "Distributed Consensus in Bounded-degree Networks", Colloquium, T.J. Watson I.B.M. Research Center (July 1990).
- G. Schnitger, "Sigmoids versus Binary Thresholds", Invited Talk, DIMACS Workshop on Lower Bounds, Princeton University (Jan. 1991).
- I. Parberry, "On the Dynamic Range of Synaptic Weights in a Discrete Neuron Model", Invited Talk, Special Session of The American Mathematical Society Meeting on Mathematical Issues in Biologically Motivated Computing, The University of South Florida, Tampa, FL (March 1991).
- P. Berman, "Improved Approximation for Rectilinear Steiner Trees", Colloquium, AT&T Bell Laboratory, Murray Hill (April 1991).
- P. Berman, "Improved Approximation for Rectilinear Steiner Trees", Colloquium, T.J. Watson I.B.M. Research Center (May 1991).
- G. Schnitger, "Sigmoids versus Binary Thresholds", Colloquium, University of Dortmund (June 1991).
- W. Maass, G. Schnitger and E. D. Sontag, "On the Computational Power of Sigmoid versus Boolean Threshold Circuits", 32nd Annual Symp. on Foundations of Computer Science, (To be Presented in Oct. 1991).

References

- [1] P. Berman and J. Garay. Optimal early stopping in distributed consensus. Research Report RC-16746, IBM Research Division, T.J. Watson Research Center. July 1990.
- [2] P. Berman and J. Garay. Cloture votes: n -resilient, polynomial time distributed consensus in $t + 1$ rounds. *Mathematical Systems Theory*. To Appear.
- [3] P. Berman and J. A. Garay. Asymptotically optimal distributed consensus. *Proc. 16th International Colloquium on Automata, Languages, and Programming*, pages 80-94. 1989.
- [4] P. Berman, J. A. Garay, and K. Perry. Toward optimal distributed consensus. *Proc. 30th Ann. IEEE Symp. on Foundations of Computer Science*, pages 340-346. 1989.

- [5] P. Berman, I. Parberry, and G. Schnitger. The complexity of reliability and constraint satisfaction in neural networks. *Neural Networks*, 1(Supplement 1):76, 1988.
- [6] P. Berman, I. Parberry, and G. Schnitger. A note on the complexity of reliability in neural networks. Technical Report CRPDC-91-3, Center for Research in Parallel and Distributed Computing, University of North Texas, 1991.
- [7] P. Berman and A. Pelc. Distributed probabilistic fault diagnosis for multiprocessor systems. *Proc. of the 20th Annual International Symposium on Fault-Tolerant Computing*, pages 340-346, 1990.
- [8] P. Berman and B. Roos. Learning one-counter languages in polynomial time. *Proc. 28th Ann. IEEE Symp. on Foundations of Computer Science*, pages 61-67, October 1987.
- [9] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. *Proc. 6th Annual Symposium on Theoretical Aspects of Computer Science*, February 1989.
- [10] J. Chu and G. Schnitger. The communication complexity of several problems in matrix computation. *Proc. 1st ACM Symposium on Parallel Algorithms and Architectures*, pages 22-31, 1989.
- [11] B. Kalyanasundaram and G. Schnitger. Rounds versus time for the two person pebble game. *Proc. 6th Annual Symposium on Theoretical Aspects of Computer Science*, pages 517-529, 1989.
- [12] P. Berman M. Obradovic and Z. Obradovic. Comparison of learning models for boolean functions. Technical Report CS-88-21, Dept. of Computer Science, Penn State University. 1988.
- [13] M. Obradovic and P. Berman. Voting as the optimal static pessimistic scheme for managing replicated data. *Proceedings of the Ninth Symposium on Reliable Distributed Systems*, pages 126-135, 1990.
- [14] M. Obradovic and P. Berman. Weighted voting for operation dependent management of replicated data. *Proceedings of the Fourth Workshop on Distributed Algorithms*, Oct. 1990.
- [15] Z. Obradovic and I. Parberry. *Analog Neural Networks of Limited Precision I: Computing with Multilinear Threshold Functions*, pages 702-709. Morgan Kaufmann, 1990.
- [16] Z. Obradovic and I. Parberry. Learning with discrete multi-valued neurons. *Proceedings of the Seventh Annual Machine Learning Conference*, pages 392-399, 1990.

- [17] Z. Obradovic and I. Parberry. Learning with discrete multi-valued neurons. *Journal of Computer and System Sciences*, To Appear.
- [18] I. Parberry. A primer on the complexity theory of neural networks. In R. Banerji, editor, *Formal Techniques in Artificial Intelligence: A Sourcebook*, volume 6 of *Studies in Computer Science and Artificial Intelligence*, pages 217-268. North-Holland, 1990.
- [19] I. Parberry and G. Schnitger. Relating Boltzmann machines to conventional models of computation. *Neural Networks*, 2(1):59-67, 1989.
- [20] G. Schnitger W. Maass and E. D. Sontag. On the computational power of sigmoid versus boolean threshold circuits. *Proc. 32nd Annual Symp. on Foundations of Computer Science*, 1991, To Appear.
- [21] P. Y. Yan and I. Parberry. Exponential size lower bounds for some depth three circuits. Technical Report CRPDC-90-1, Center for Research in Parallel and Distributed Computing, Department of Computer Sciences, University of North Texas, 1990.